INTRODUCTION

The aim of this project is to predict the salary of a Job by looking at the content of the Job description and some other data related to the job. This problem was first proposed in Kaggle by the company Adzuna that wanted to build a prediction engine for the salary of any UK based job advertisement (https://www.kaggle.com/c/job-salary-prediction). This is basically a regression problem where we have to predict the salary based on the inputs from the features of the dataset. The features includes the job description, location, contract details, company name and the type of job. The minimum error of the winning project in Kaggle was 3464.55 and in our project we aim to attain an error less than that(or atleast close enough). We also aim to apply clustering to do some knowledge discovery on the dataset.

Data Format and Preprocessing

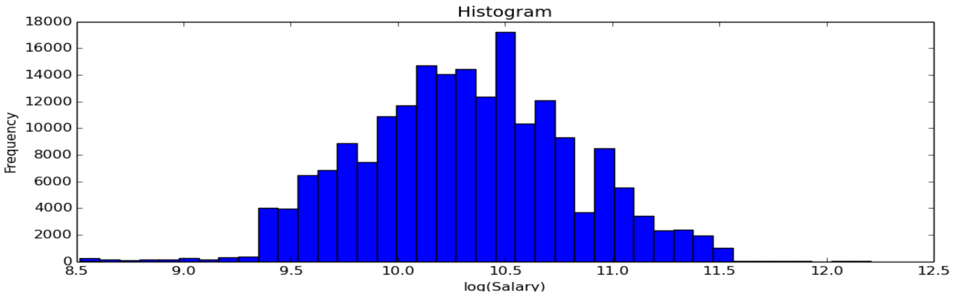
The data used in this project has 7 features - Category, Company, Contract Time, Full Description, Location and Title. The value that has to be predicted is the salary.

For preprocessing we had to convert each string feature into a vector. We could have either used Count Vectorizer[2] of Tf-Idf vectorizer[1]. The problem with Count Vectorizer is that it gives a higher weight to bigger documents since they have more words. We used Term Frequency(TF) and Inverse Document Frequency(IDF) to convert string in our feature set into a vector of fractions. TF-IDF gives low weight to words that are more frequent across documents and high weight to words that are less frequent across documents.

For 'full description' feature, we used 10000 as size of the word vector , whereas for every other feature we used a vector of size 1000. So in total our data matrix size was 2,40,000 x 18,732. We then divide into 80 - 20 , where we use 80% to train the model and test it on 20%. From training model we use 5-fold cross validation to get the best value of the magic parameters of the model by using bias-variance tradeoff.

We apply classification algorithms on our dataset , even though it is a regression problem. We do this because we would like to first classify the data-set and then accordingly apply regression in each of the classified cluster. For that we first plot the distribution of the salary bucket. We then try to make the distribution normal. We take log(salary) and the curve resembles normal curve. After that we divide the dataset into three buckets based on the standard deviation of the normal curve. We change the salary distribution to Y={0,1,2} depending on the value of Y, and use the new dataset as a classification problem.

The diagram below shows the distribution of log(salary)



Once we classify the data-set , we can do regression on each bucket to get the final salary value.

Data Models

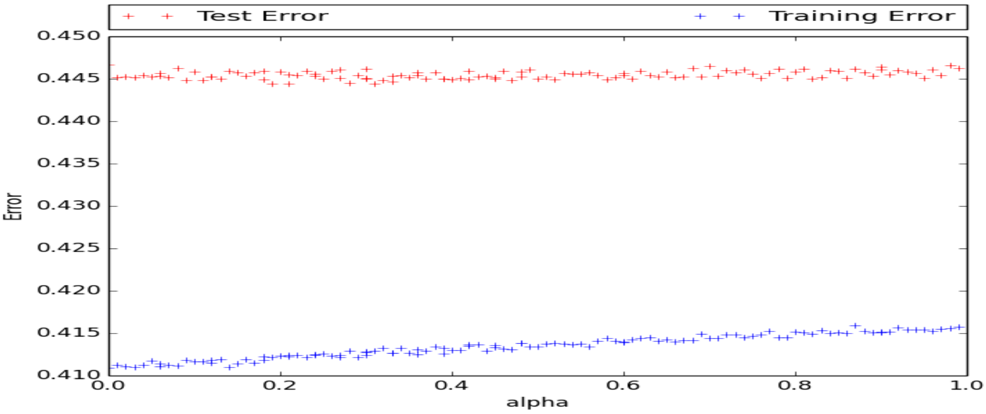
Our approach was to first apply linear data models (MNB[3] , Logistic Regression[4]) and to see how the data fits into the model. If the dataset does not do well, we can assume that our dataset is not linearly separable and we would need a model that finds non-linear boundaries. Unfortunately, our data-sets failed to perform well for linear classification. For non-linear boundaries first we tried Decision Tree classifier and Regression[5]. Then, we tried the meta-algorithm Adaboost using MNB(Multinomial Naive Bayes) to find non-linear boundaries in dataset.

We also applied K-Means algorithm in the dataset and tried to analyze the best clusters formed. Finally, we tried to do knowledge discovery by analyzing the clusters.

Multi Nomial Naive Bayes Algorithm[3]

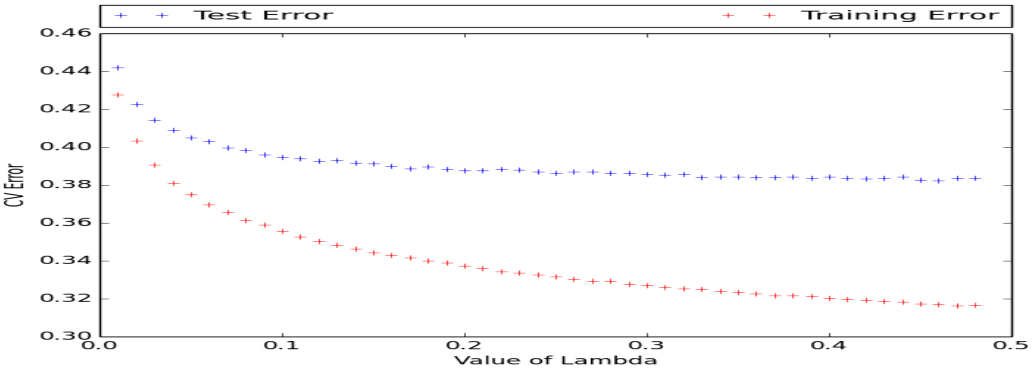
This algorithm is extremely popular for document classification and so we use this as the first model that we apply on our dataset. The regularization parameter 'alpha' is calculated using 5-fold cross validation.

If we see the below diagram, the training error starts with 0.410 at regularization parameter 0.0 and keeps increasing with increasing alpha , whereas the test error remains constant at 0.45. We weren't able to analyze much from this result. The algorithm wasn't able to classify our dataset correctly. We tried from 0-4 value of alpha in step interval of 0.01.



Logistic Regression [4]

Since MNB makes assumption of conditional independence, we tried an algorithm that doesn't make any assumption about the dataset.
The regularization parameter 'lambda' was inferred using 5-fold cross validation.



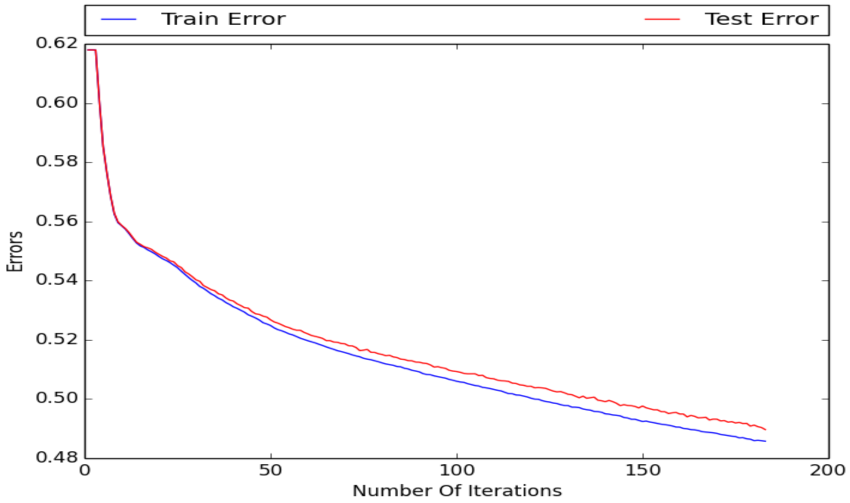
Logistic regression gives better results than MNB. The training error decreases from 0.45 to 0.31 , whereas the test error is almost constant at 0.40 with increasing alpha. This algorithm gives better results than MNB, but its still not good enough.

Since the two linear classifiers does not give good results we tried non-linear classifiers hoping to get better results.

Adaboost with MNB[6]

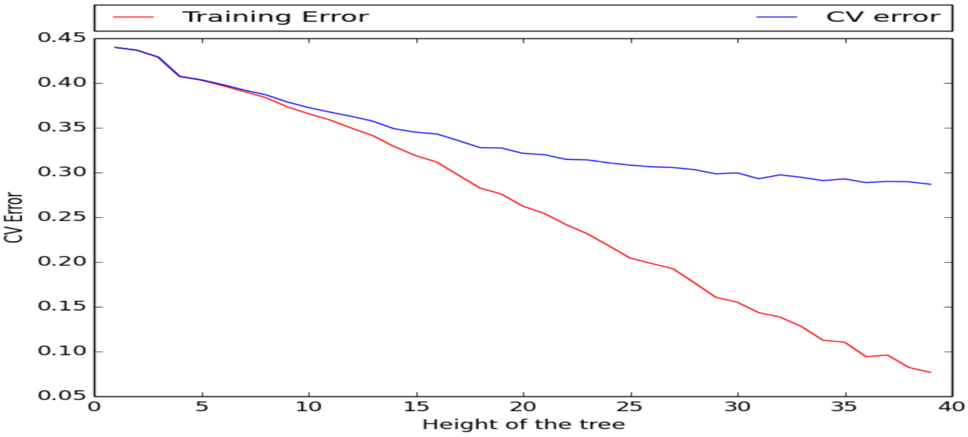
MNB didn't work on our dataset , but we thought we can use MNB with Adaboost to find nonlinear boundaries in our dataset. Also MNB runs really fast and is scalable and with increasing number of iterations in Adaboost, MNB converges faster than LR or Decision Tree.

But unlike of what we thought , Adaboost did not give the results we expected. We tried till 200 iterations of Adaboost , but we still could not get error below 0.43.



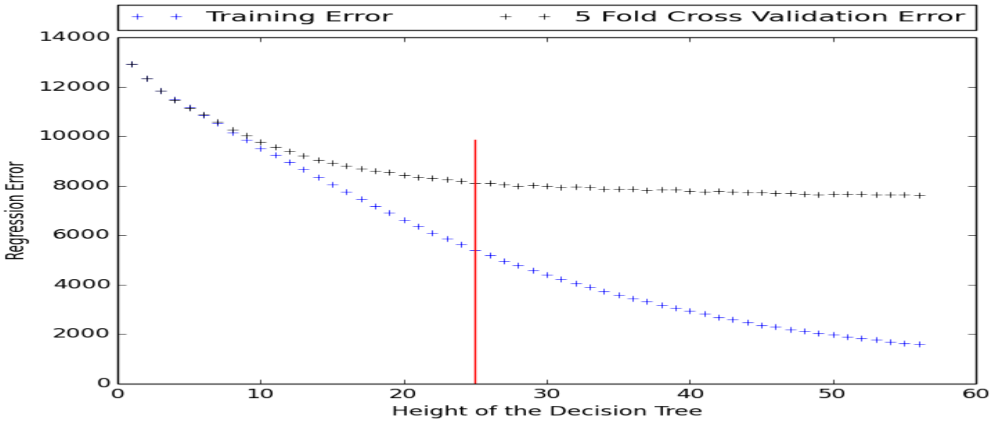
Decision Tree Regression[5]

We tried Decision Tree classifier, using 5 fold cross validation. We got the best error value of 0.27 at tree with height 27. We used 5-fold cross validation to calculate the height of the tree.



The training error goes to 90% at height 40, and the test error is almost constant at 0.27 after height 27. Training error accuracy of 90% at height 40 shows that the model is over fitting at that height.

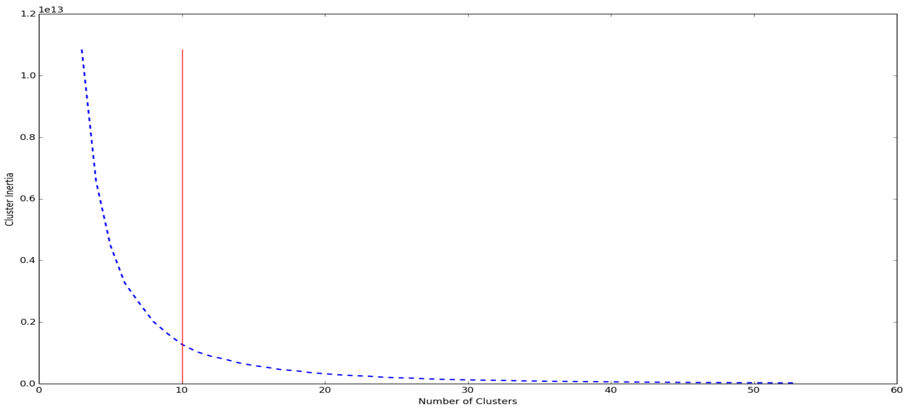
We use Decision Tree regression and try to find the best height by plotting regression error and not classification error. We got height of 26 as the ideal height using 5-fold cross validation.



These are all the algorithms that we have tried till now for classifying the data-set. We tried applying SVM on our dataset but it did not work(and did not converge) because of the size of our dataset.

K-Means

We also tried to do some knowledge mining by clustering the dataset. We applied K-means clustering on the dataset. The distance measure used in this algorithm is Euclidean distance.



As we can see in the above graph, the best cluster size is 10. We cluster the data into 10 clusters and below are description of each cluster.

Clusters	**Size**	Min Salary	Max Salary	Unique Job Source	Number of Unique Cities
0	46260	24610	31171	jobs.cabincrew.com	2574
1	7694	68371	82875		287
2	20315	46713	56500		931
3	40056	10000	9999	cvjobstore.com	2546
4	28763	38395	46674		1561
5	239	120000	200000		12
6	5190	100000	99999		238
7	13797	56560	68256		621
8	39306	31180	38379		2139
9	43148	18132	24604	Personneltoday Jobs, Jobcentre Plus , jobsinrisk.com	3243

The cluster 5, is the cluster that has the maximum salary in it. From analyzing that cluster we see that there are 12 cities, which contains jobs with salary only in that range. These entire cities are present in UK with 4 of them in Manchester, 3 in Liverpool and 5 in British Virgin Islands.

The highest pay is received by Job titles that contain words like analyst, doctor, trading, equity and business.
Also the website jobs.cabincrew.com sends job emails only in the salary range of cluster 0 i.e 24-31K.
Similarly Personnel Today jobs and jobinrisk.com offers jobs with salary between 18-24K.

Future Work

Before the final project presentation we aim to try few more stuffs to see if we can get better results.
Since we have divided dataset into clusters, we would like to train regression algorithm for each cluster and use that to get final salary. We will first try to assign a new data point to a cluster and then predict the salary using regression algorithm trained using data in each cluster.
We also aim to do density estimation on the data to find out the probability distribution of the features and find outliers in the data. We would then see whether we could extract some useful knowledge using the above method.
We also aim to apply spectral clustering using Rbf kernel. This should cluster data different than K-means and we may be able to get some other interesting information about the data.

References

[1]Scikit TfIdf Transformer http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html
[2] Count Vectorizer - http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[3] C.D. Manning, P. Raghavan and H. Schuetze (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265. http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
[4 Logistic Regression using]LIBLINEAR – A Library for Large Linear Classification http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, Belmont, CA, 1984.

[6] Y. Freund, R. Schapire, “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting”, 1995